REPORT 35c3ctf\_junior

.1996

buffer overflow

Redirect to spawnshell offset 0x418

.poet

:v bài này em bof bằng biến poem trong hàm rate\_poem xong redirect về reward

Hoặc là mình overflow author để write trên cái điểm = 100000

.stringmaster1

Bug: không check string[npos]-> overwrite đc string\_length. Khi print mình sẽ leak được stack. (Do string <16 byte nên còn nằm trên stack)

Trong hàm replace() em nhập 1 char không có trong str1 và replace 1 char bất kì, xong print nó sẽ leak stack.

Em thấy được return address của hàm Play() nằm trên stack. Xong em đặt address của spawn\_shell lên stack = replace rồi swap nó với ret của Play -> Khi Play ret sẽ jump tới spawn\_shell.

.Stringmaster2

Bug: cũng giống như stringmaster1

Nhưng mà bài này nó không cho mình shell như bài trước + ASLR bật nên em nghĩ sẽ là ret2libc

Đầu tiên em sẽ leak libc\_base

Xem qua thử stack = gdb thì thấy trên stack có \_\_libc\_start\_main+231 ở offset 0x88 và ret address của hàm Play() ở offset 0x78

Cũng như bài trước, mình leak stack tính được libc\_base.

Tới đây em cũng hong biết là return về system xong cat flag được không :v tại làm local hoặc là return về one\_gadget trong libc.

.Sum

Bug: bài này mới vào nó read vào 1 biến size\_t mà không check <0 nên mình int overflow thành max\_size\_t

+ Calloc(-1) ret NULL mà nó cũng không check nên mình sẽ control được cái pointer get với set

Em thấy nó tính chỗ set hoặc get = r12+8\*rax (r12 ret của hàm calloc, rax là index do mình nhập). Mà r12=0 khi calloc(-1,)+ index read vào là usign\_int64 nên mình có thể arbitrary write hoặc read trên heap.

Ý tưởng là em sẽ leak libc xong r cũng overwrite hàm nào đó để call system

Bài này có cho hàm puts nên sẽ leak libc =puts

Em gọi get puts\_got/8 -> ra được puts trên GOT, tính được libc\_base. Xong em set address của free\_got thành system, tại khi mình bye nó sẽ call free() -> chắc v là ra shell xong cat flag